Git and GitHub

TA: Ziqi Yang

Today’s goals

- By the end of today, you should be able to…
  - describe what version control systems are and their purposes
  - push and pull remote repositories from local computers
  - version control local files using git clients

A Quick Notice

- This is a discussion session - we are here to help guide you through the course materials
- Please feel free to interrupt us with questions or comments!

Agenda

- VCS (Version Control Systems)
- What is Git and GitHub
- Features
- Git Client
- Basic use
- Branches and GitFlow (if time permits)

Agenda

- VCS (Version Control Systems)
- What is Git and GitHub
- Features
- Git Client
- Basic use
- Branches and GitFlow (if time permits)

Why Version Control?

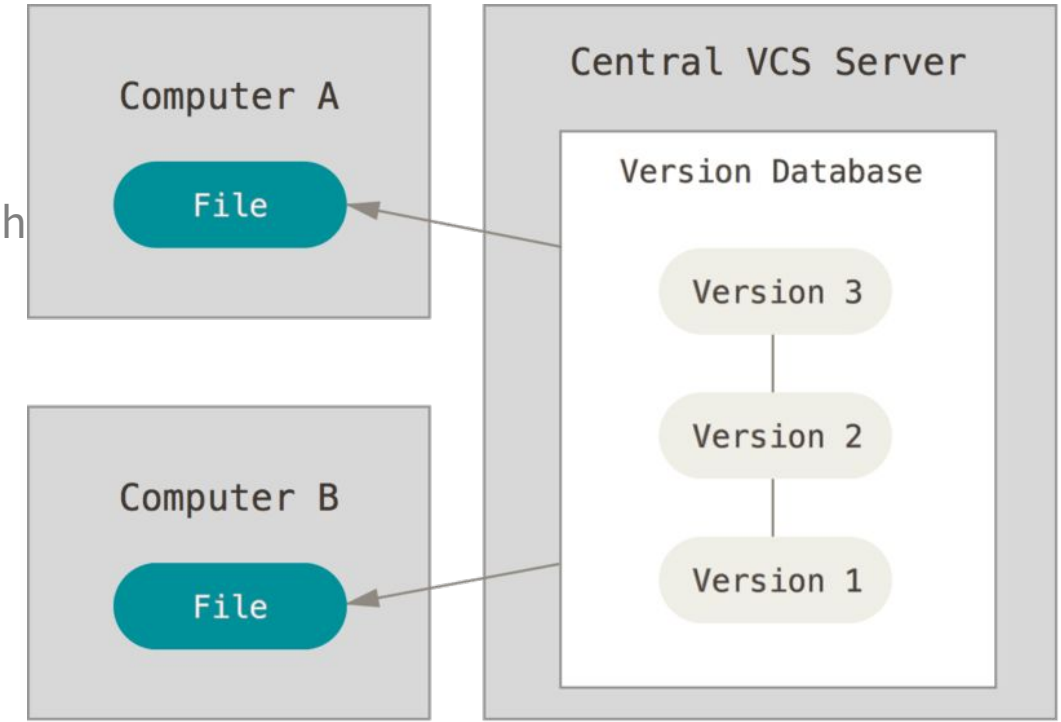
Source:
https://twitter.com/aksharpathak

VCS (Version Control Systems)

- Enable collaboration between many developers
- Recover files or revert to previous state
- Identify who made modifications/issues
- Two main types
    - Centralized VCS
    - Decentralized VCS

VCS (Version Control Systems)

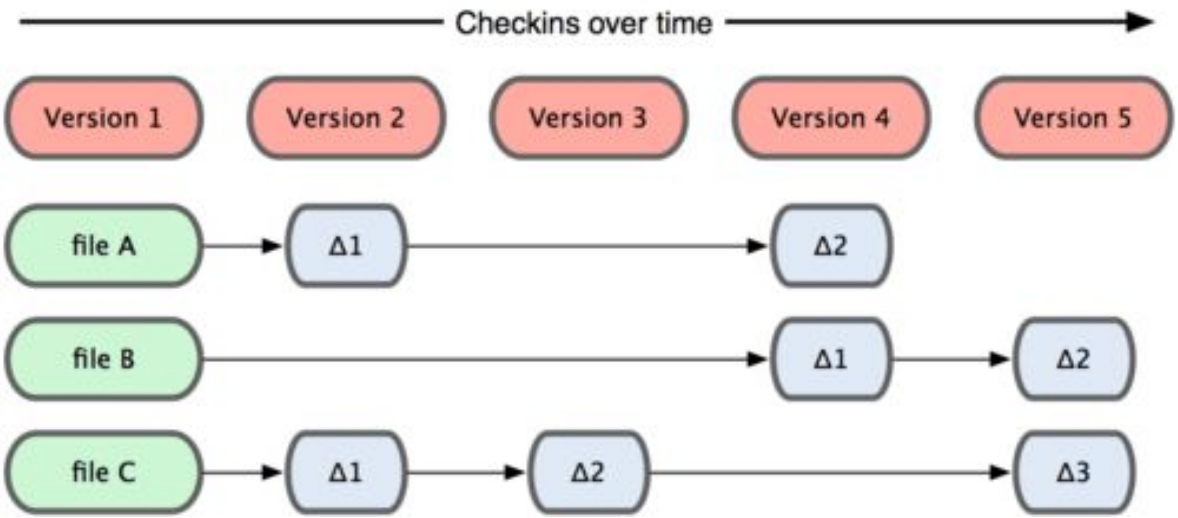
- Centralized VCS
  - Central Repository
  - Limited and high cost branch
  - Ex: SVN



Source:
https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

VCS (Version Control Systems)

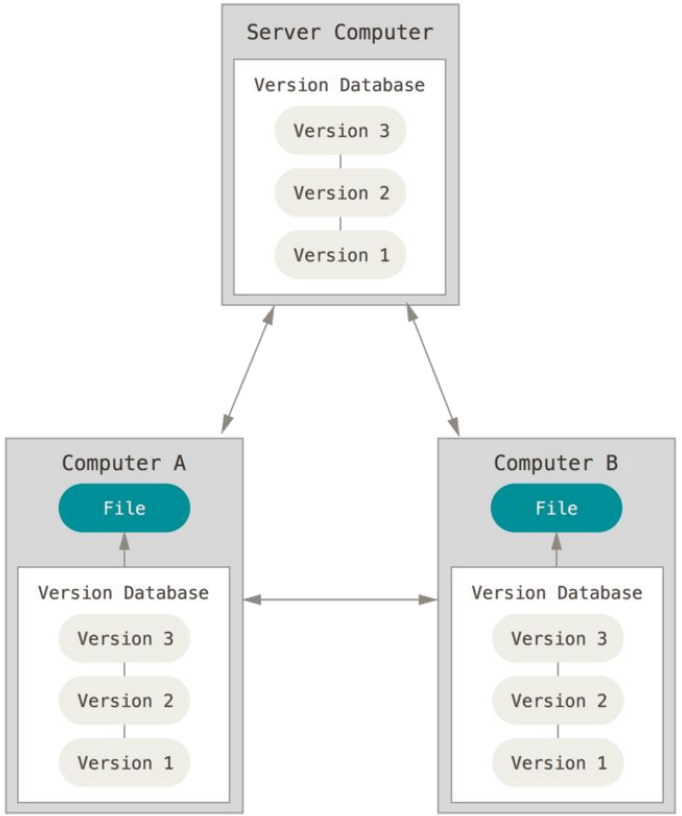
- Centralized VCS
  - SVN: Stores list of changes to files



Source: https://git-scm.com/book/en/v1/Getting-Started-About-Version-Control

VCS (Version Control Systems)

- Decentralized VCS
  - Each collaborator has a copy
  - E.g.: Git

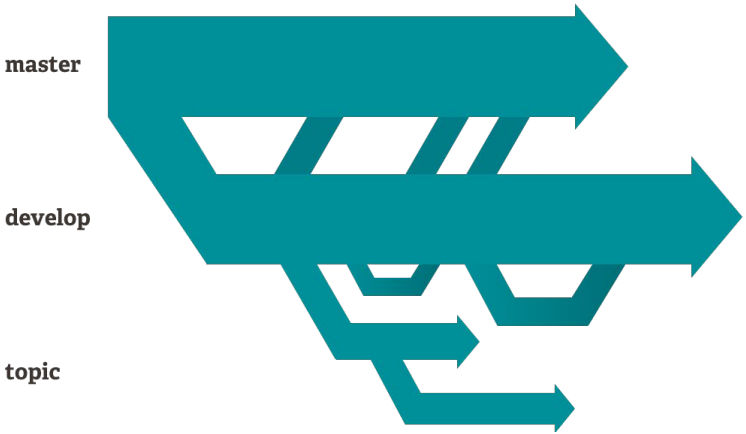


Source:
https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Git and GitHub

Git:

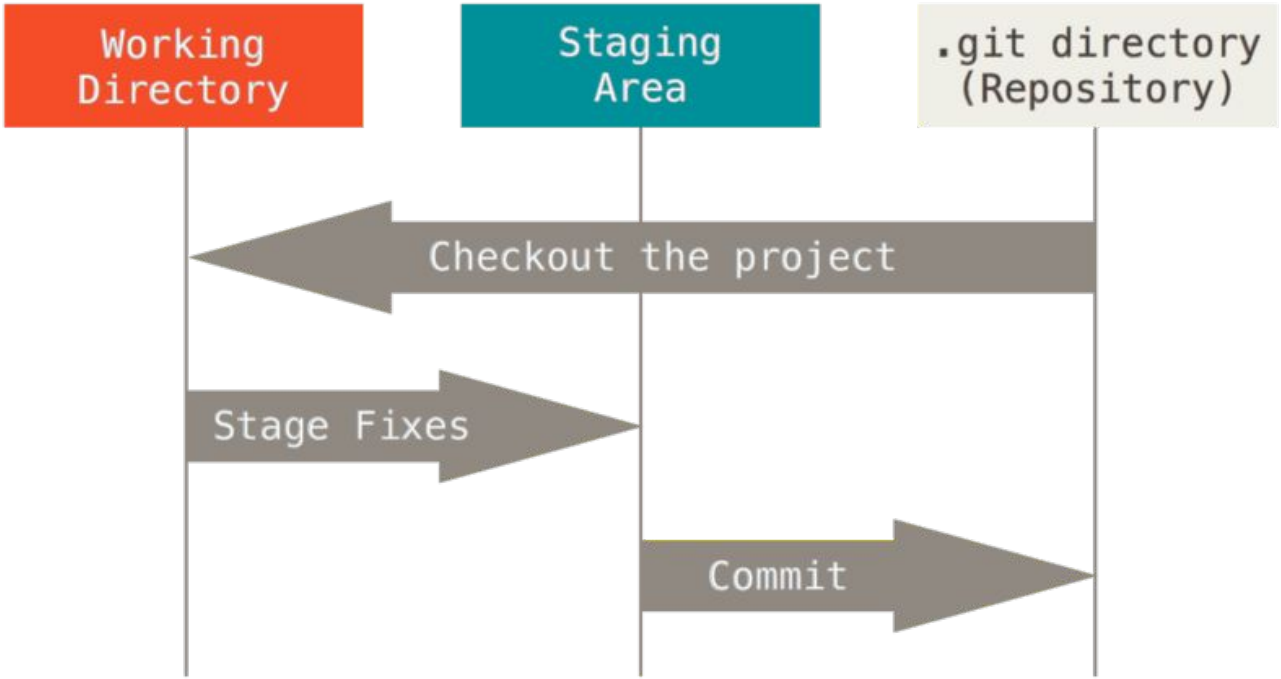
- Free and open source system
- Focused on integrity, speed and high collaboration
- Nearly every operation is local
- Decentralized VCS



Git and GitHub

Git **local** file states:

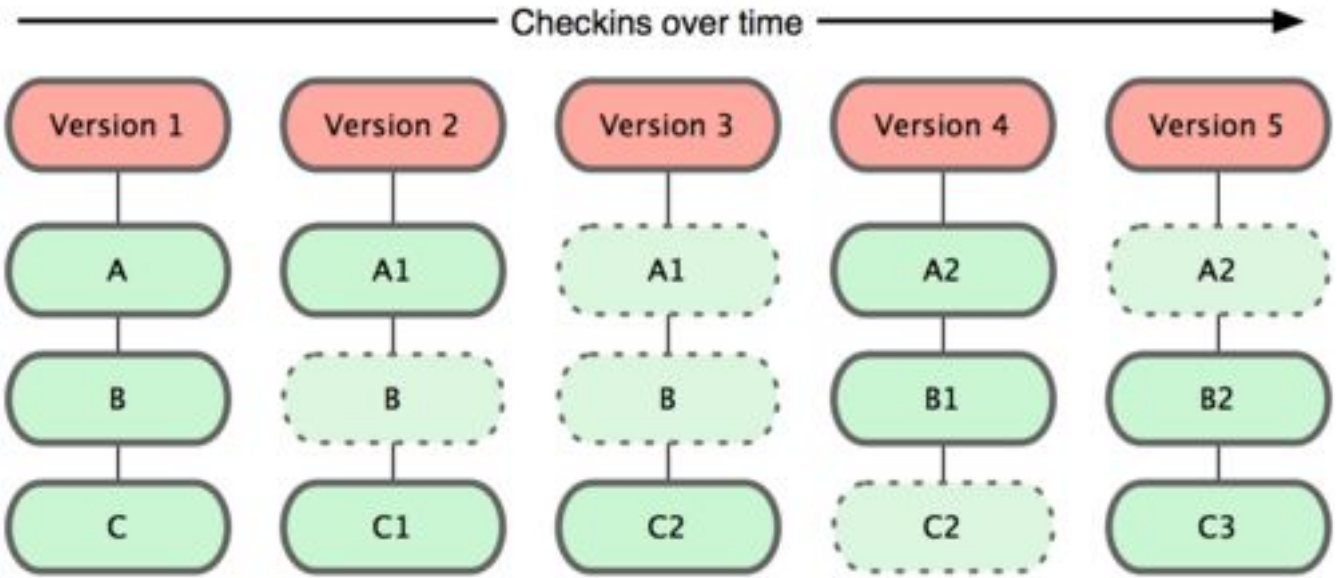
- Modified
- Staged
- Committed



Source:
https://git-scm.com/book/en/v2/Getting-Started-What-is-Git%3F

Git and GitHub

Git stores data as snapshots



Source:
https://git-scm.com/book/en/v1/Getting-Started-About-Version-Control

Git and GitHub

GitHub: service for hosting Git repository.

- Git is the tool, GitHub is the service to host projects that use Git.
- Other options: Bitbucket, GitLab, and many others.
- Mostly free with other premium plans
- Large community collaboration on open source projects
    - (Good for your CV/Portfolio)

Features

- Reliable storage of your data
- Public or private repositories (based on your plans)
- Collaboration between different people
- Version control
- Repository search (programing language, framework, etc)
- Repository ranking (stars)
- Addons/Plugins

Agenda

- VCS (Version Control Systems)
- What is Git and GitHub
- Features
- Git Client
- Basic use
- Branches and GitFlow (if time permits)

Git Clients

- There are many GUI tools
  - SourceTree
  - GitKraken
  - Github Desktop
- For this discussion, we will cover Github Desktop and Command Line Interface
  - Installation
  - Basic Usage
- I (and some other TAs) personally use CLI much more :)

An example or Assignment 1

Using Git Command Line

Installing Git Command Line

- Install using your favorite package manager
  - https://git-scm.com/download/linux
  - For example:

```
# apt-get install git               // Debian-based
# yum install git                   // Fedora (up to 21)
# pacman -S git                     // Arch
```

- I personally like
  - HomeBrew: https://brew.sh/
- An official guide: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Configuring Git

- Configure git with your email and name
  - These are NOT credentials for your server
  - Local records that show who made which commit

```
$ git config --global user.name “Peter the Anteater”
$ git config --global user.email “peteranteater@uci.edu”
```

Using Git Command Line

- Initializing a new local repository
  - Navigate to a directory you want to start version controlling and initialize

```
you@local:~$ cd /your/local/project
you@local:/your/local/project$ git init
```

- Commit changes (local)
  - Modified files need to be staged (tracked) for commit by using git add

```
you@local:/your/local/project$ touch new_file.txt
you@local:/your/local/project$ git add new_file.txt
you@local:/your/local/project$ git commit -m “Added new_file.txt”
```

Using Git Command Line

- Push
  - Set the remote url to your newly-created github repository

```
you@local:/project$ git remote add origin https://github.com/your-repo.git
```

  - Push to your github repository

```
you@local:/project$ git push origin master
```

  - You need to push in order to save the changes in your hosted repository (e.g., github)

Using Git Command Line

- Fetch, Merge
  - Fetch - a “safe” way to download changes to your local repository
    - Does not merge changes with local

```
you@local:/project$ git fetch origin
```

  - Merge - applies the remote’s changes with local

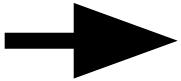
```
you@local:/project$ git merge origin
```

- Pull
  - Identical to fetch + merge

```
you@local:/project$ git pull origin
```

Using Git Command Line

- Conflict when pulling
  - Merge conflict error



```
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/jonghl9/example
 * branch            master     -> FETCH_HEAD
   739a077..c46bbee  master     -> origin/master
Auto-merging my-file.txt
CONFLICT (content): Merge conflict in my-file.txt
Automatic merge failed; fix conflicts and then commit the result.
```

  - Resolve conflict, add and commit file

```
1 This is a line Bob wrote in his local computer.
2 This is a second line Bob wrote in his local computer.
3 This is a line Bob wrote in his local computer.
```

Original

Conflict

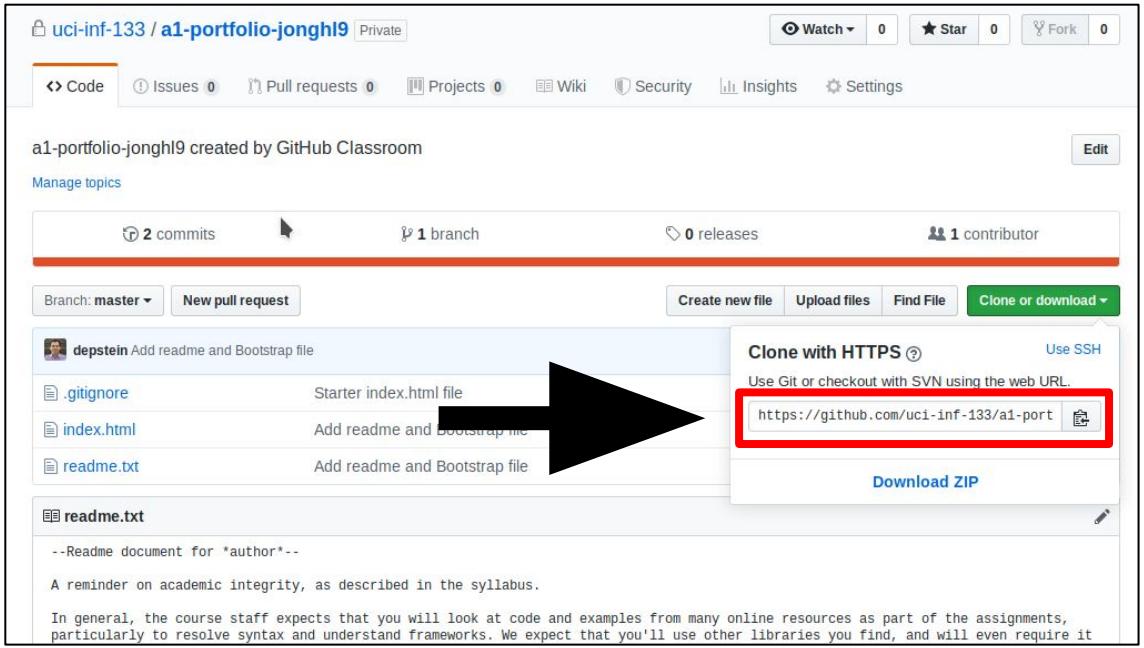
```
1 This is a line Bob wrote in his local computer.
2 This is a second line Bob wrote in his local computer.
3 <<<<<<< HEAD
4 This is a line Bob wrote in his local computer.
5 =======
6 This is a line Bob wrote in his remote.
7 >>>>>>> c46bbee024fe3da6f9608df1020413bfafe054ce
```

Resolved

```
1 This is a line Bob wrote in his local computer.
2 This is a second line Bob wrote in his local computer.
3 This is a a line Bob wrote to resolve the conflict.
```

Using Git Command Line

- Downloading an existing repository from Github



Using Git Command Line

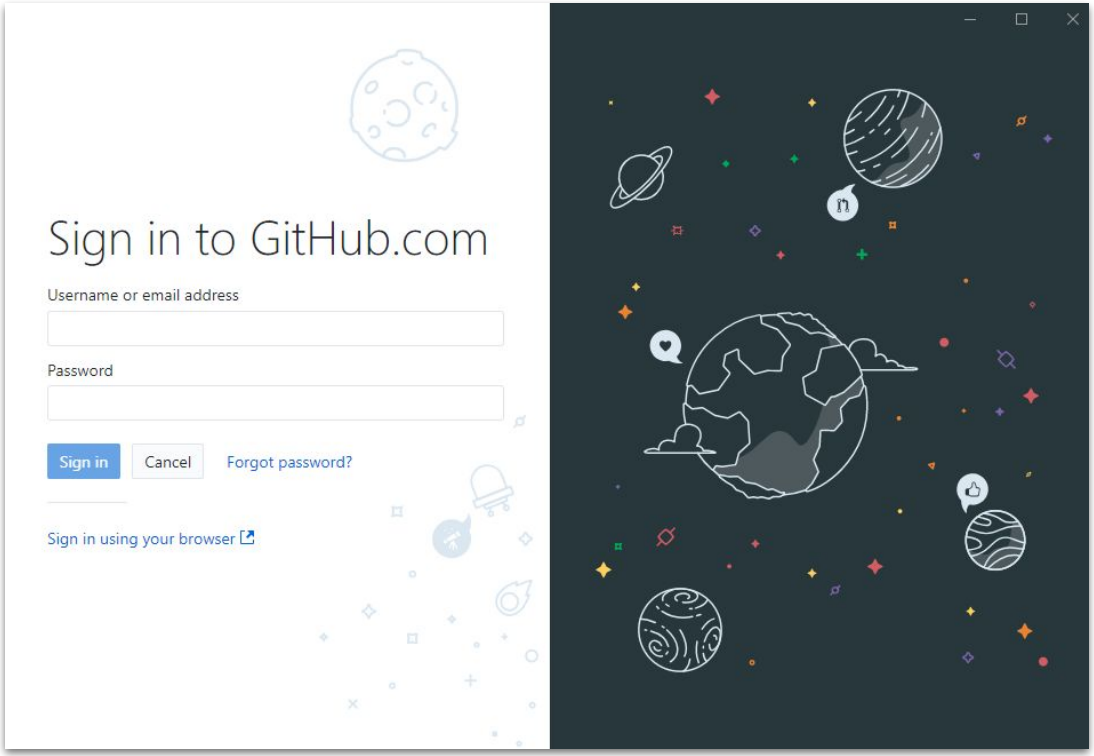
- Downloading an existing repository from Github

```
you@local:~$ git clone https://github.com/your/github/repo.git
```

Using GitHub Desktop

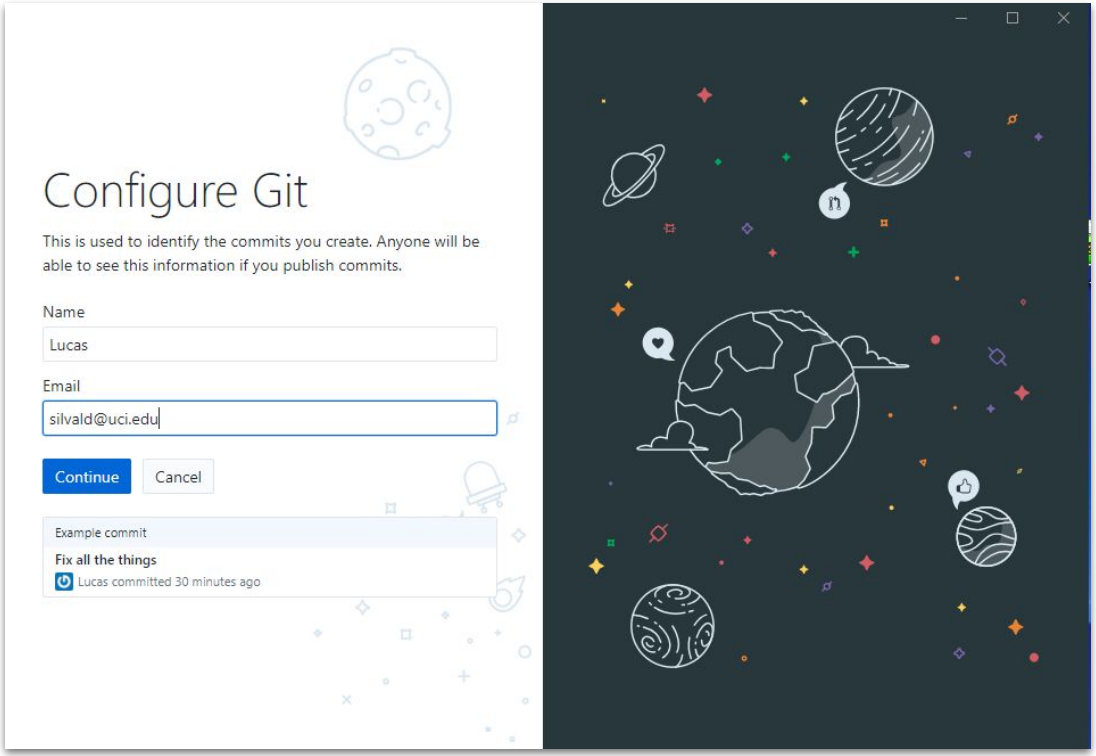
Installing Desktop Client

1. Sign up on github.com
2. Download client at https://desktop.github.com (Mac OS and Windows)



Installing Desktop Client

Configure Git



Agenda

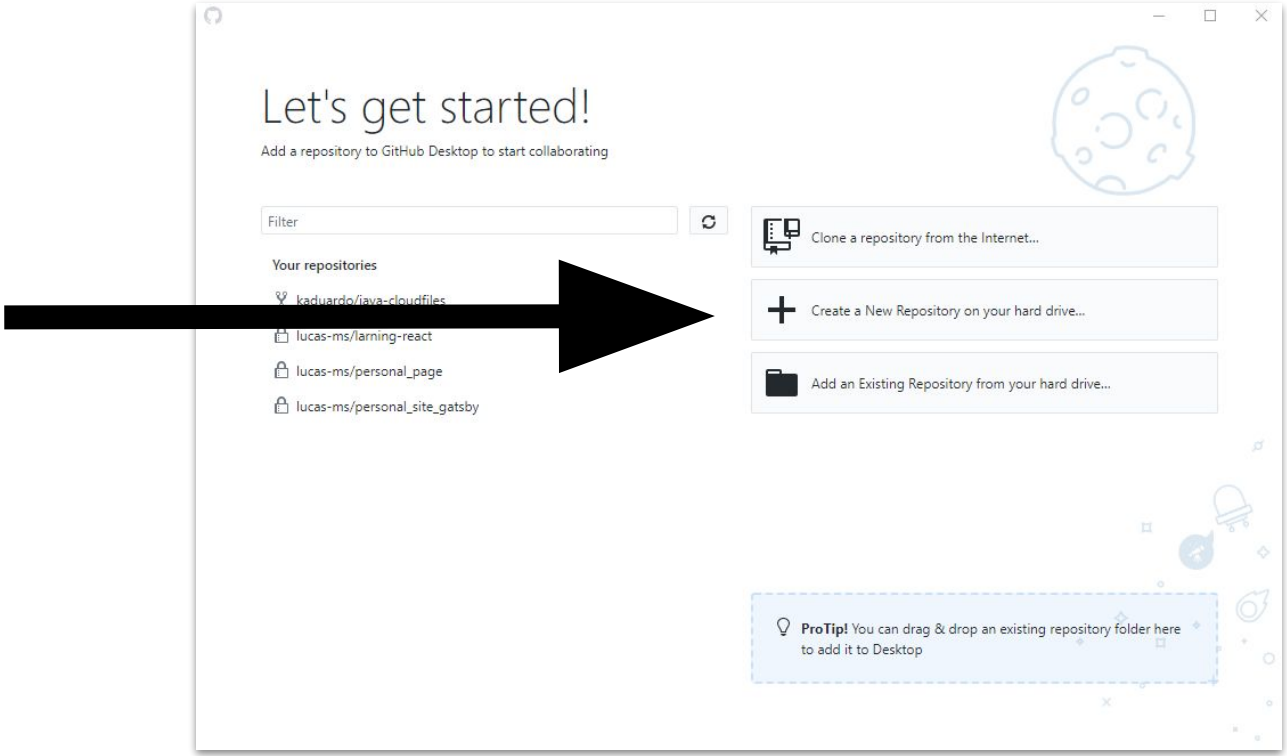
- VCS (Version Control Systems)
- What is Git and GitHub
- Features
- Git Client
- Basic use
- Branches and GitFlow (if time permits)

Basic Use

- New repository
- Commit
- Push/Publish
- Fetch, merge / pull
- Clone repository

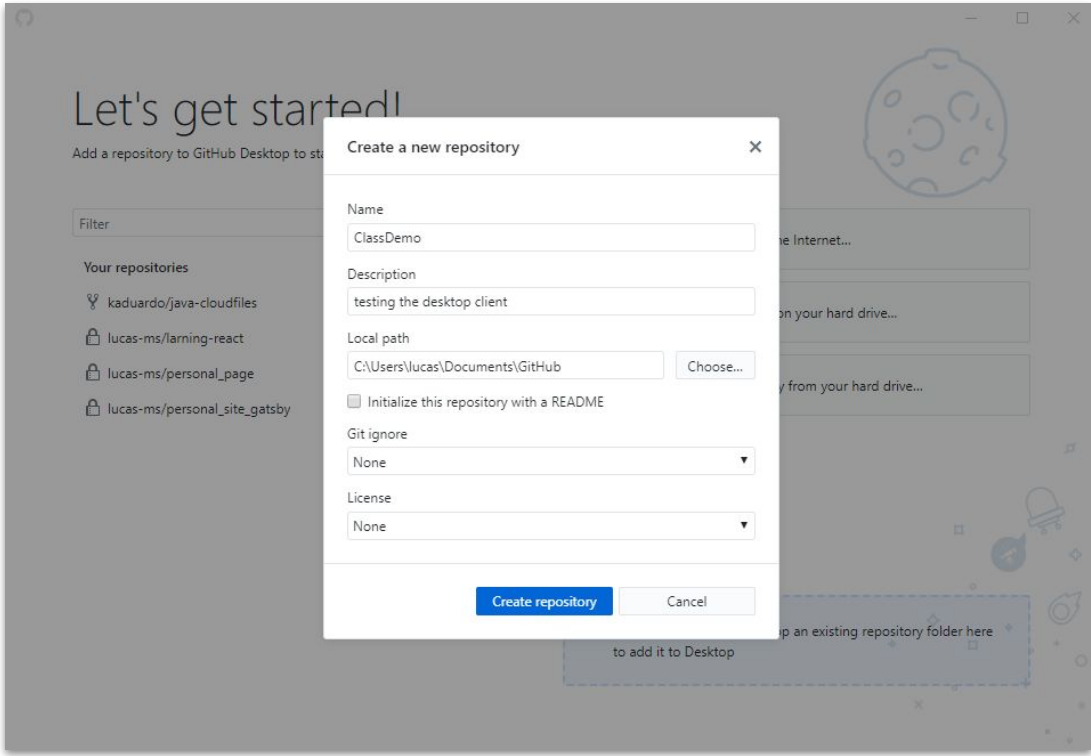
Using Desktop Client

New Repository



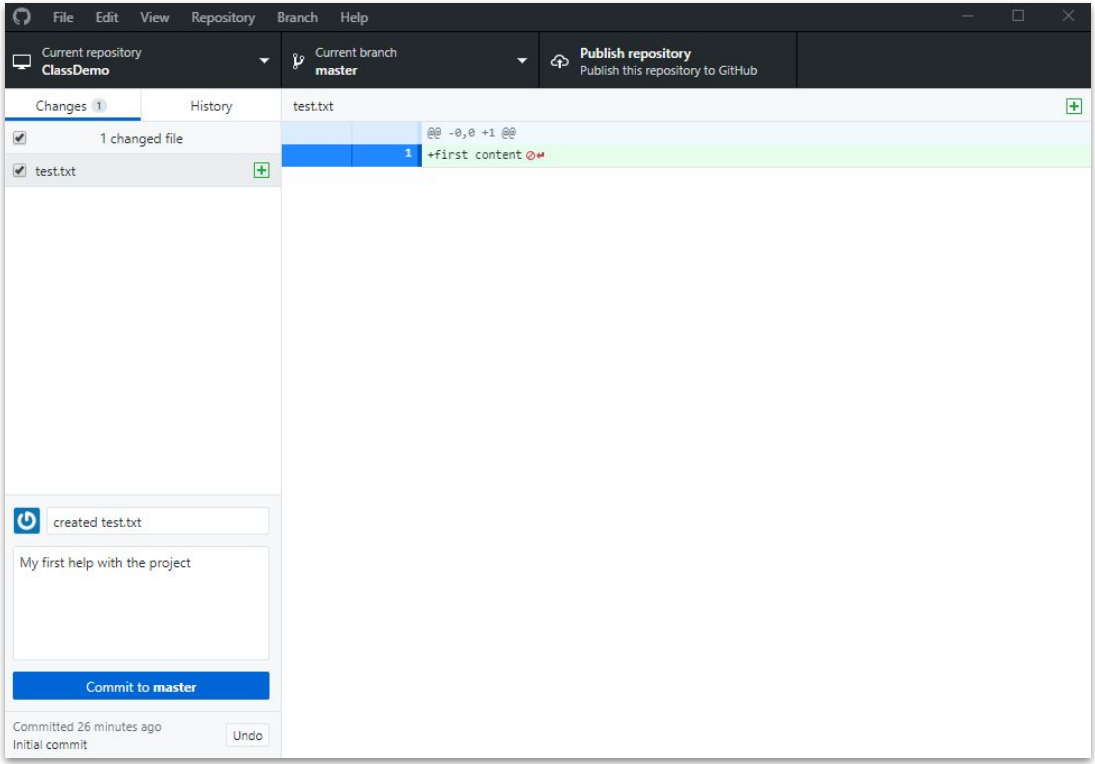
Using Desktop Client

New Repository



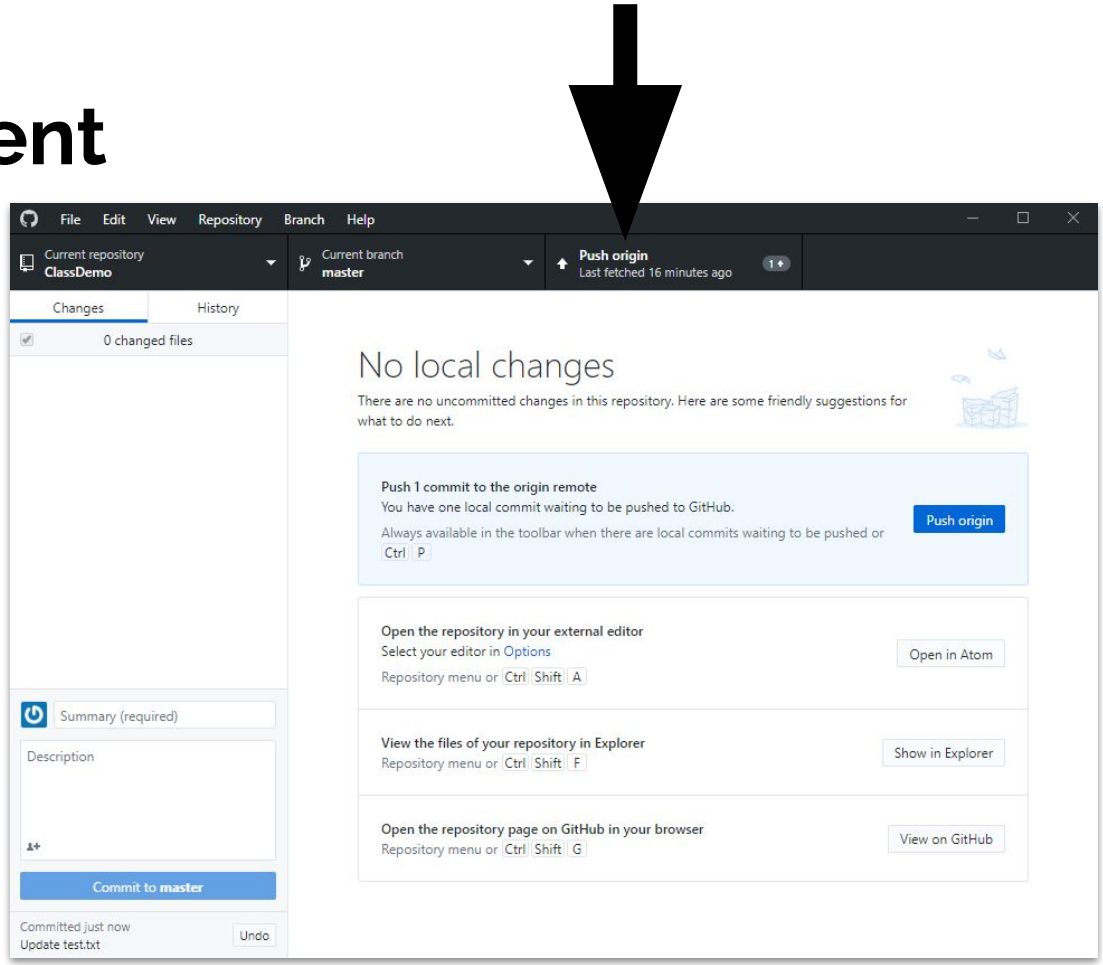
Using Desktop Client

- First commit
  - Still only local!
- Make as many commits as necessary



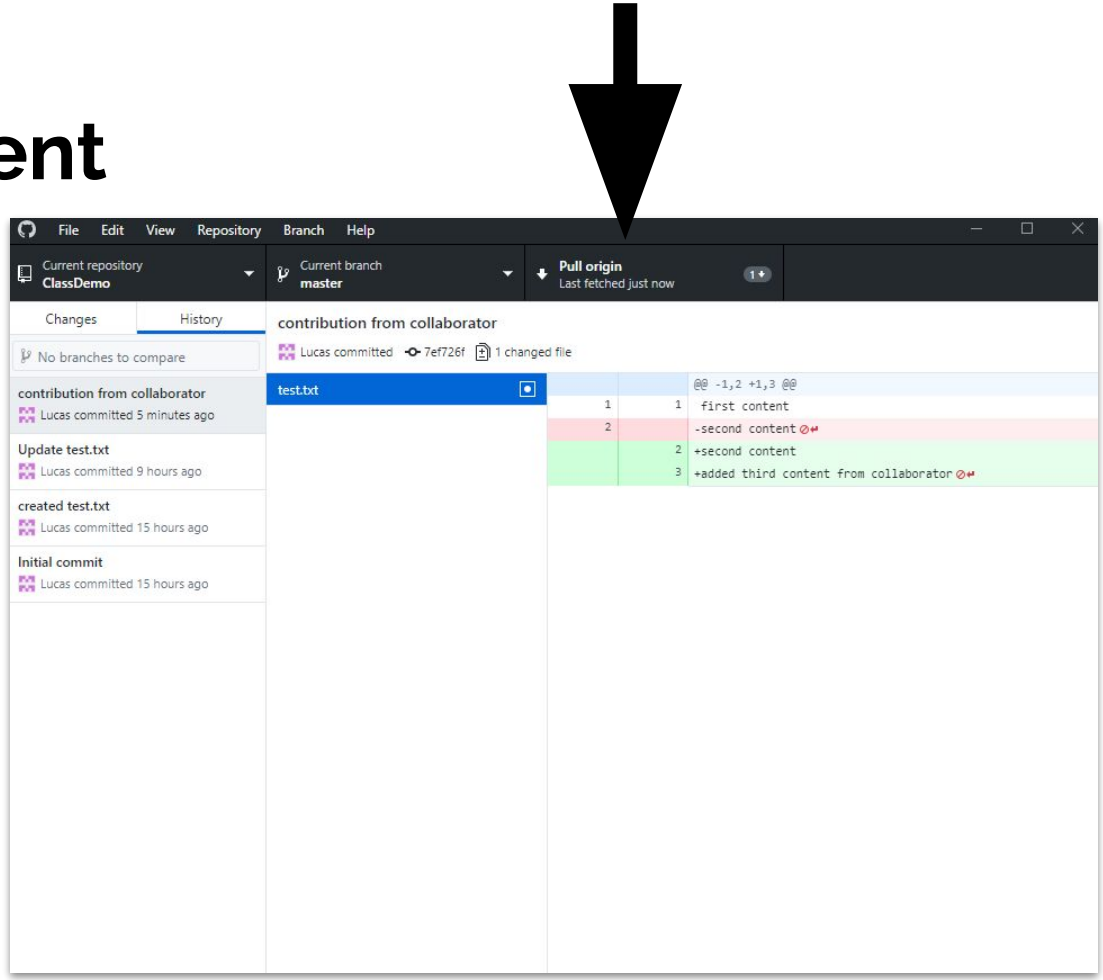
Using Desktop Client

- Push
  - Now sends all local commits to server



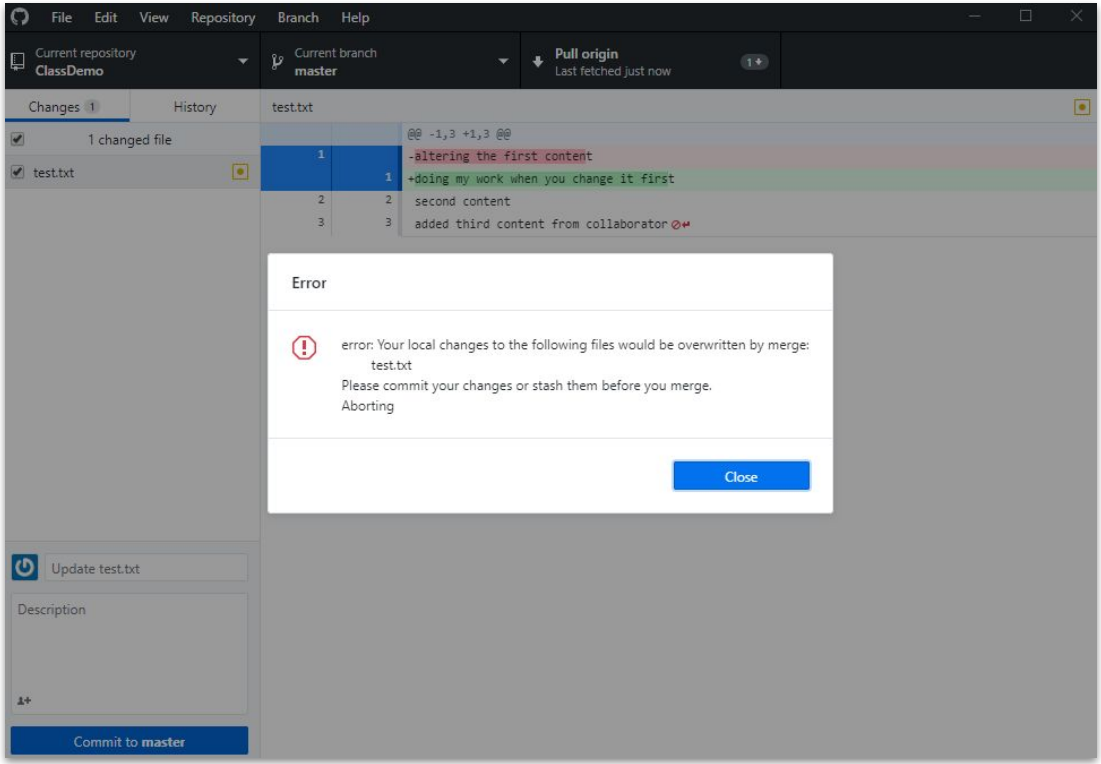
Using Desktop Client

- Fetch
  - Downloads data from server
- Pull
  - Downloads + Merges



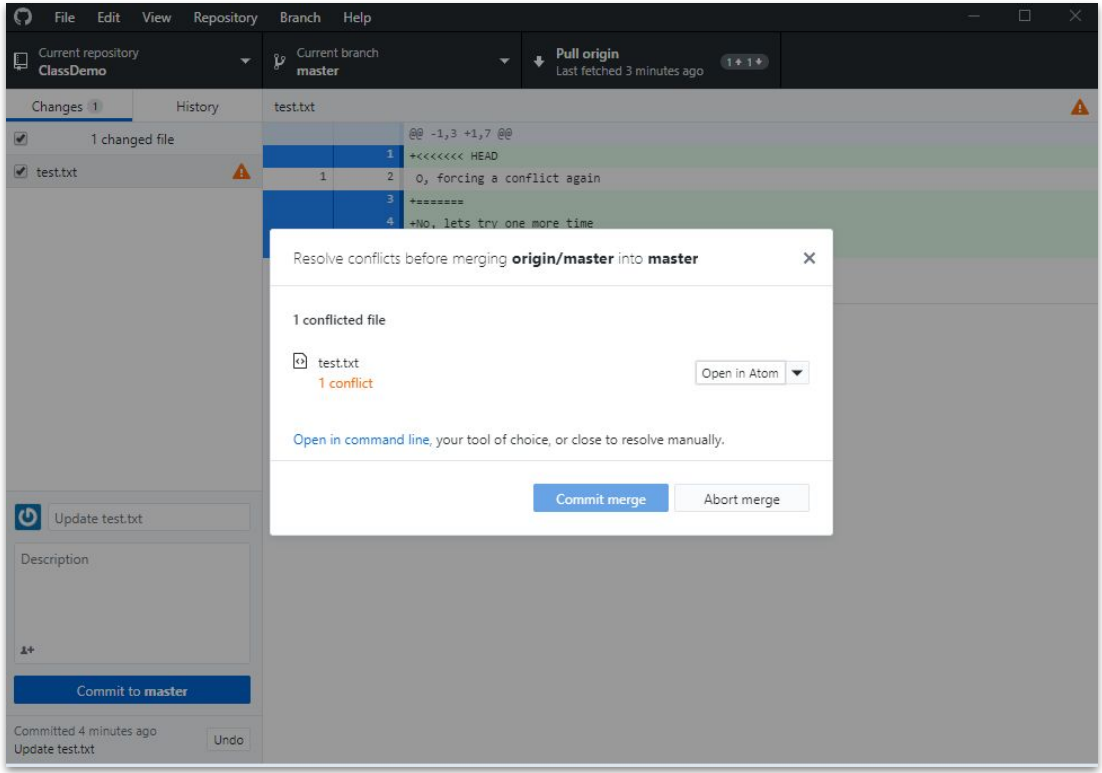
Using Desktop Client

- Conflict when pulling (merging)



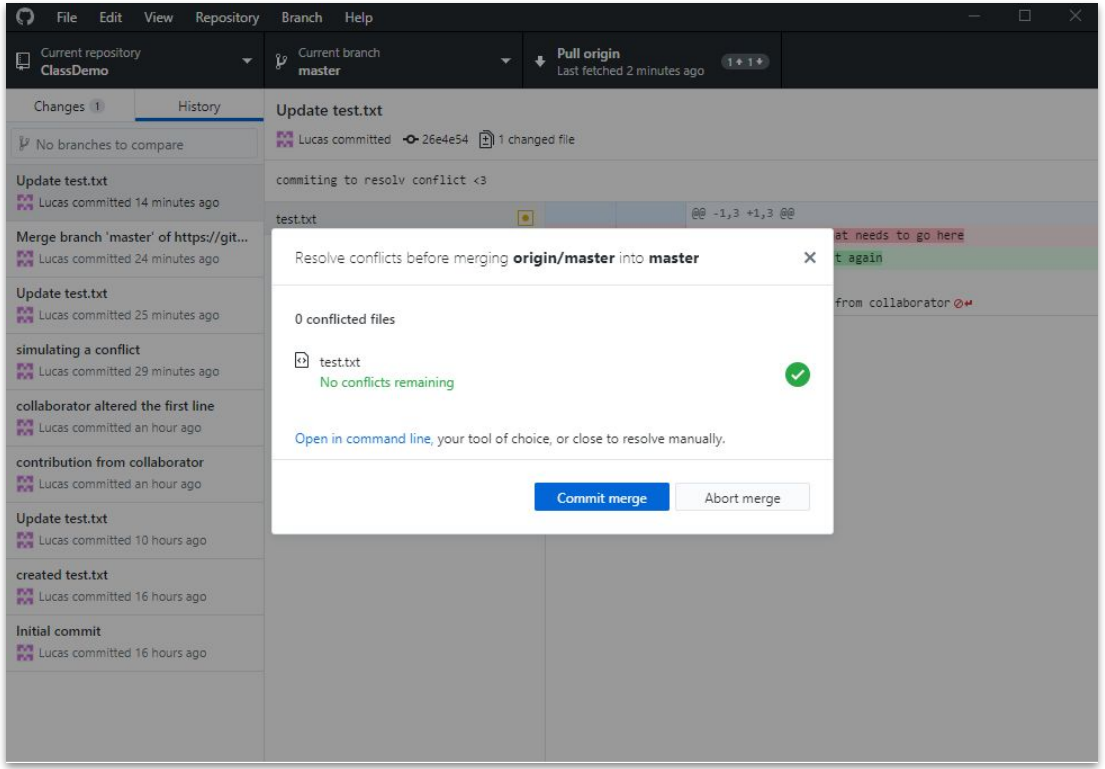
Using Desktop Client

- Conflict when pulling (merging)
  - Commit your changes to local repository



Using Desktop Client

- Conflict when pulling (merging)
  - Commit your changes to local repository
  - Resolve conflict
  - Commit merge
  - Push



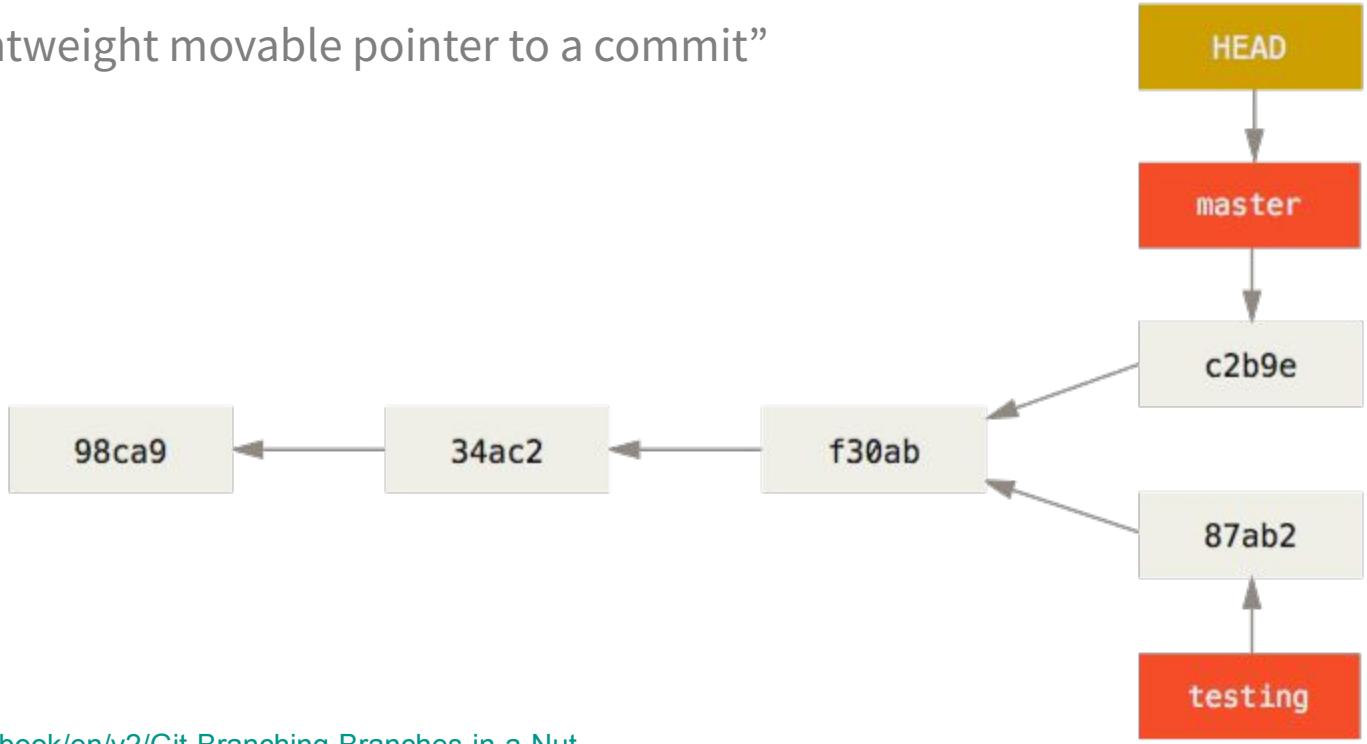
Lets try it out now!

Agenda

- VCS (Version Control Systems)
- What is Git and GitHub
- Features
- Git Client
- Basic use
- Branches and GitFlow

Branches

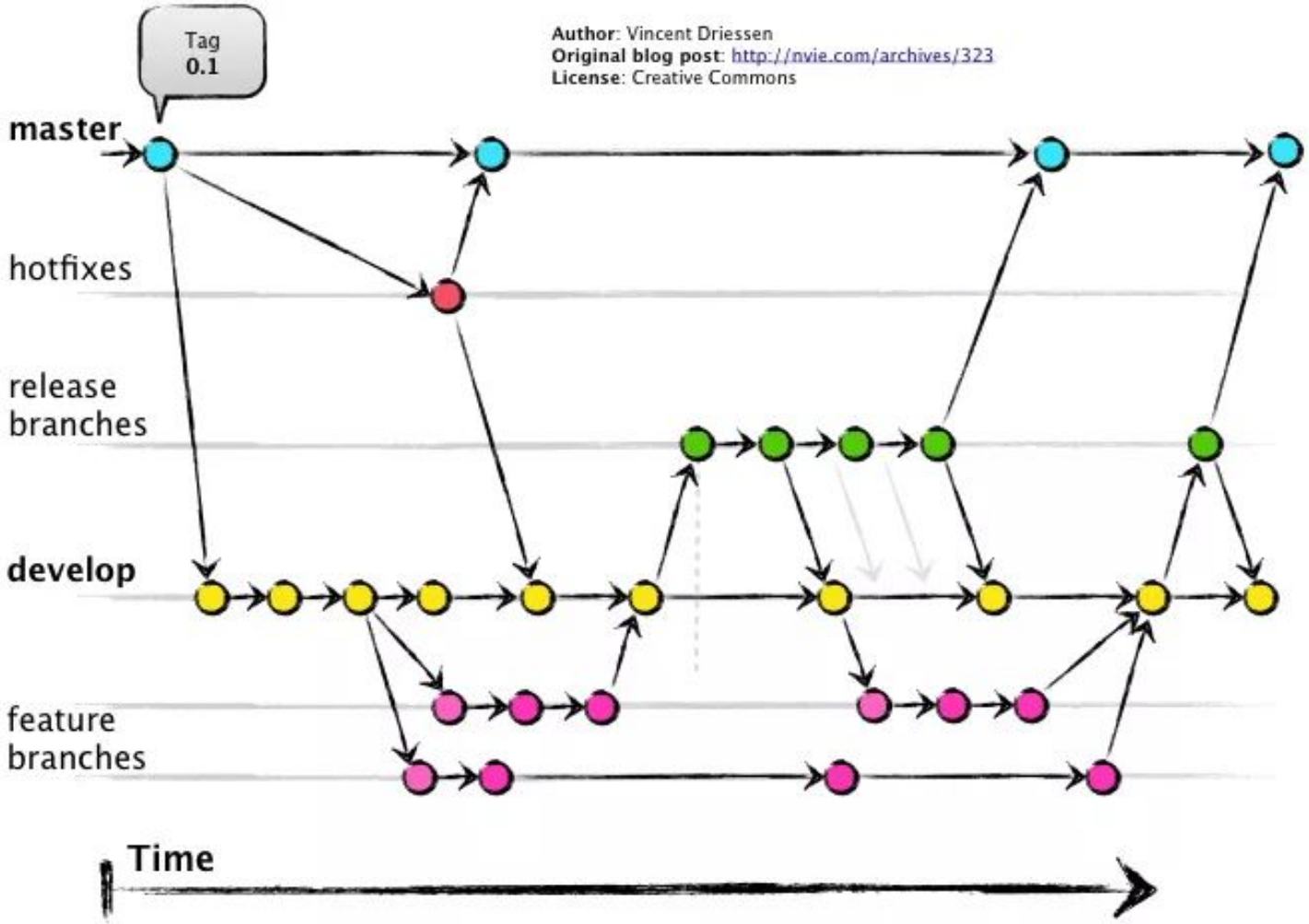
- “A lightweight movable pointer to a commit”



Source:
https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell

GitFlow

- A workflow methodology
  - Organizes bug fixes, releases, features, etc
  - Directs collaboration in large projects


Tag
0.1
Author: Vincent Driessen
Original blog post: http://nvie.com/archives/323
License: Creative Commons
master
hotfixes
release
branches
develop
feature
branches
Time

Further Resources

- Atlassian git tutorial
  - https://www.atlassian.com/git/tutorials
- Git documentation
  - https://git-scm.com/docs
- GitFlow
  - GitFlow https://nvie.com/posts/a-successful-git-branching-model/
- GitHub Flow
  - https://guides.github.com/introduction/flow/
- My TA Office hours: See course website
- Credits to previous TAs Lucas and Jong for the slides